

“RUPIYA”

A PROJECT REPORT



**Government College Of Commerce and
Business Administration
sector-50, Chandigarh**

AFFILIATED TO:

PANJAB UNIVERSITY, CHANDIGARH



For qualifying BCA Degree
2023-24

Submitted to:

*GCCBA Faculty
(Dr. Deepli)*

Submitted by:

*Sidharath
2133014/21
BCA 6th sem*

CERTIFICATE

This is to certify that this project (Rupiya) has been completed by

1. Sidharath, 2133014/21, BCA 6th semster
2. Aditya Kumar Singh, 2133053/21, BCA 6th semester

We hereby certify that this project is the original work of us and no part has been copied from any mainstream source.

This project is completed under the guidance of Ms. Deepali and the project is completed as per the guidelines of the university.

This project submission is a partial fulfillment of the requirement for qualifying the BCA degree under Panjab University, Chandigarh.

TEAM MEMBER DETAILS:

- 1) Sidharath, 2133014/21, BCA-6th semester
- 2) Aditya Kumar Singh, 2133053/21, BCA-6th semester

This project has been prepared exclusively for academic purposes, hence it shall not be used for any kind of non-academic purpose whatsoever.

Internal Signature

External Signature

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide Ms.Deepali as well as our principal who gave me the golden opportunity to do this wonderful project on the topic “Rupiya”, which also helped me in doing a lot of research and i came to know about so many new things I am really thankful to them.

I am also grateful to our mentors and advisors for their valuable guidance, insightful feedback and unwavering support. Their mentorship has been instrumental in steering me in the right direction and helping me overcome challenges along the way.

TABLE OF CONTENT:

S.No.	Topic	Page no.
1	Introduction to project	
2	Features, benefits, objectives	
3	Hardware and software specification of “Rupiya”	
4	Feasibility study	
5	Modular description(Design behavior)	
6	Flow diagram of working of “Rupiya”	
7	Code of the project	
8	Testing and implementation	
9	Snapshot of project working	
10	Conclusion	
11	Reference	

INTRODUCTION

Managing Your Finances Effortlessly with RUPIYA

In today's fast-paced world, managing personal finances efficiently is crucial for individuals to achieve their financial goals and secure their future. Recognizing this need, RUPIYA presents itself as a reliable solution, empowering users to take control of their financial journey effortlessly.

The integration of technology has revolutionized traditional approaches, offering innovative solutions that cater to the evolving needs of users. Among these solutions stands "RupiYa" a web-based application meticulously crafted to empower individuals in managing their finances effectively. Developed with a blend of PHP, HTML, CSS, and JavaScript, RupiYa emerges as a versatile platform, seamlessly blending functionality with user-centric design.

RupiYa's primary objective is to provide users with a centralized hub for tracking income, monitoring expenses, and gaining valuable insights into their financial behaviors. By leveraging modern web technologies, the application ensures accessibility across devices, enabling users to stay connected to their financial data anytime, anywhere. This accessibility aligns with the contemporary lifestyle, where individuals seek convenience and flexibility in managing their finances amidst their busy schedules.

At the heart of RupiYa lies its ability to simplify financial tracking. Through dynamic message sliders, users are introduced to key features such as responsible savings, expense management, and insight full data visualization. These messages encapsulate the core ethos of RupiYa, encouraging users to embrace financial prudence and avoid overspending.

As we embark on this journey through RupiYa, envision a future where financial management is no longer a daunting task but a fulfilling endeavor. Together, let us embrace the power of RupiYa to transform the way we perceive and manage our finances, one click at a time.

FEATURES-

1.User Authentication and Security: “Rupiya” ensures the security of user accounts and financial data through robust authentication mechanisms, such as password encryption and session management. This feature safeguards sensitive information and prevents unauthorized access to user accounts.

2. Dashboard Overview : Upon logging in, users are greeted with a comprehensive dashboard providing an overview of their financial status. This includes summaries of income, expenses, savings goals, and any upcoming financial commitments. The dashboard serves as a central hub for accessing various features and functionalities of the application.

3.Expense Tracking: Rupiya allows users to track their expenses effortlessly. Users can categorize expenses, add descriptions, and specify payment methods for each transaction. This feature provides insights into spending patterns and helps users identify areas where they can potentially save money.

4.Income Management: Users can record their sources of income, including salaries, freelance earnings, investments, and any other sources. Rupiya enables users to categorize income streams and track their earnings over time. This feature facilitates better budgeting and financial planning.

5.Budgeting Tools: Rupiya offers budgeting tools that enable users to set financial goals and allocate funds accordingly. Users can create personalized budgets for different categories, such as groceries, utilities, entertainment, and more. The application provides alerts and notifications to help users stay on track with their budgeting goals.

6.Financial Reporting : Rupiya generates detailed financial reports, including income statements, expense summaries, and balance sheets. Users can customize reports based on specific time periods, categories, or criteria. These reports offer valuable insights into financial trends and patterns, empowering users to make informed decisions.

7.Goal Setting and Tracking: Users can set savings goals, debt repayment targets, or investment milestones within the application. Rupiya tracks progress towards these goals and provides visual representations of achievement. This feature motivates users to stay focused on their financial objectives and celebrate milestones along the way.

8.Interactive Charts and Graphs: Rupiya utilizes interactive visualizations, such as charts and graphs, to present financial data in a clear and engaging manner. Users

can explore their financial information visually, making it easier to understand trends, patterns, and correlations within their finances.

BENEFITS-

1.Enhanced Financial Awareness : Rupiya provides users with a holistic view of their financial situation, including income, expenses, and savings goals. By offering detailed insights and analysis, the application promotes greater awareness of spending habits and financial behaviors.Users gain a deeper understanding of where their money is going, enabling them to make more informed decisions regarding budgeting and saving.

2.Improved Budgeting and Planning : With its budgeting tools and goal-setting features, Rupiya empowers users to take control of their finances and plan for the future effectively. By setting realistic budgets and tracking expenses, users can identify areas for potential savings and allocate funds towards specific financial goals. This proactive approach to budgeting helps users achieve financial stability and progress towards their long-term objectives.

3.Streamlined Financial Tracking : Rupiya simplifies the process of tracking income and expenses, eliminating the need for manual record-keeping and reconciliation. Through automated categorization and transaction management, users can easily monitor their financial transactions in real-time. This streamlined approach saves time and effort, allowing users to focus on more strategic aspects of financial management.

4.Increased Financial Accountability: By providing users with a centralized platform for managing their finances, Rupiya fosters a sense of accountability and responsibility. Users are encouraged to stay on track with their budgeting goals and savings targets, knowing that their financial data is securely stored and readily accessible. This accountability motivates users to make prudent financial decisions and avoid unnecessary expenditures.

OBJECTIVE-

The primary objective of Rupiya is to empower individuals with effective tools and resources for managing their finances efficiently. Through its user-friendly interface and comprehensive features, Rupiya aims to promote financial awareness,facilitate prudent decision-making, and ultimately enhance users' financial well-being.

The application seeks to streamline the process of tracking income, monitoring expenses, and setting financial goals, providing users with a clear understanding of their financial standing and enabling them to take control of their financial futures. By offering budgeting tools, goal-setting functionalities,and detailed financial reports, Rupiya empowers users to make informed decisions about budget allocation, expenditure prioritization, and long-term savings strategies.

Moreover, Rupiya prioritizes security and accessibility, ensuring that users' financial data remains protected and readily accessible across various devices. By fostering accountability, promoting financial literacy, and offering practical solutions for everyday financial management challenges, Rupiya aims to equip users with the confidence and capabilities to achieve financial stability and success.

HARDWARE AND SOFTWARE SPECIFICATIONS

1. Hardware Requirements:

- Processor: Any modern processor capable of running web browser software efficiently.
- RAM: 2GB of RAM(recommended for optimal performance).

2. Software Requirements:

Operating System:

- Windows: Windows 7 or later
- macOS: OSX10.10(Yosemite) or later
- Linux: Ubuntu14.04 or later, Debian 8 or later, openSUSE13.3 or later, Fedora Linux 24 or later

Web Browser:

- Browser (latest version recommended for best compatibility and security).

By ensuring that your system meets these hardware and software specifications, you can effectively run the website on a web browser and achieve optimal performance and compatibility.

SYSTEM ANALYSIS FOR PROJECT FEASIBILITY

Technical Feasibility:

- **Evaluation of Technology:** The project utilizes modern web development technologies such as HTML5, CSS3, JavaScript, and Node.js for server-side scripting. These technologies are widely supported and offer robust

features for building web applications.

- **Scalability:** The system is designed to be scalable, allowing for future enhancements and updates as per user requirements. The use of scalable cloud services and databases ensures that the system can handle increasing user loads efficiently.
- **Integration Capability:** The project integrates various tools and features seamlessly, ensuring compatibility across different devices and browsers. APIs and libraries are used to enhance functionality and integrate external services.
- **Data Security:** The project implements secure authentication mechanisms and encryption techniques to protect user data. Regular security audits and updates are performed to mitigate potential vulnerabilities.

Operational Feasibility:

- **User-Friendly Interface:** The system features an intuitive user interface that is easy to navigate and use, catering to users with varying levels of technical expertise.
- **Accessibility:** The project is accessible via web browsers on desktops, laptops, and mobile devices, ensuring that users can access it anytime, anywhere.
- **Maintenance and Support:** The system is designed for easy maintenance and updates. Regular backups and version control practices are implemented to minimize downtime and ensure data integrity.

Economic Feasibility:

- **Cost-Benefit Analysis:** The project offers significant benefits to users in terms of image editing and sharing capabilities, making it a valuable tool for individuals and businesses alike. The cost of development and maintenance is justified by the value provided to users.
- **Revenue Generation:** While the project is primarily focused on academic

purposes, there may be opportunities for revenue generation through sponsored content, premium features, or advertisements in the future.

- **Resource Utilization:** Efficient utilization of resources, including hardware, software, and human resources, ensures cost-effectiveness in project development and operations.

By conducting a thorough analysis of technical, operational, and economic feasibility aspects, the project demonstrates its viability and potential for success in meeting user needs and achieving its objectives

SYSTEM DESIGN AND BEHAVIOURAL DESCRIPTION

System Design:

The system design of RUPIYA , a web-based expense income manager, encompasses various components and functionalities aimed at providing users with a seamless editing experience. Below is an overview of the system design:

Frontend:

- The front-end is developed using HTML, CSS and JavaScript.
- It includes the user interface elements such as buttons, forms, and input fields.
- CSS is used for styling and responsive design, ensuring compatibility across different devices and screen sizes.

Backend:

- The backend is powered php.
- Manages server-side logic including handling user requests, processing data, and interacting with the database.
- Mysql used as the relational-database to store and manage user data, including transaction records.

Behavioral Description:

User Registration:

- The website provides a user-friendly registration form where new users can sign up by providing necessary information such as username, email address, and password.

Passwords are securely hashed using a strong hashing algorithm(e.g.,bcrypt) before being stored in the database to ensure data confidentiality and security.

Client-side validation is implemented to verify the correctness of user input(e.g.,valid email format, password strength) before submitting the registration form.

Server-side validation further validates the registration data to prevent duplicate usernames or emails and to ensure the integrity of the registration process.

User Authentication:

- Registered users can securely login to their accounts using their username/email and password combination.

Passwords are verified by comparing the hashed password stored in the database with the hashed version of the provided password during login.

Upon successful authentication, users are granted access to their personalized dashboard and financial management features.

Transaction Management:

Adding New Transactions:

- Users can add new financial transactions by specifying details such as transaction date, amount (positive for income, negative for expense), category (e.g., groceries, utilities, salary), and optional description.

Client-side validation ensures that required fields are filled out correctly, and the entered data meets specified criteria (e.g., valid date format, numeric amount).

Server-side validation processes the transaction data to ensure accuracy and consistency before inserting it into the database.

Editing and Deleting Transactions:

- Existing transactions can be edited or deleted by the user to reflect any changes or corrections.

Permissions and access controls are implemented to allow only authorized users to modify their own transactions.

Changes made to transactions are validated to maintain data integrity and consistency within the system.

DataVisualization:

Dashboard Visualization:

- Upon logging in, users are presented with a personalized dashboard that includes visual representations of their financial data.

Interactive charts and graphs (e.g., pie charts for expense categories, line graphs for income trends overtime) are generated using CanvasJS to provide users with insights into their spending habits and income patterns.

The visualization components are dynamically updated based on the user's transaction history stored in the database, ensuring real-time data representation.

PDF Export:

Exporting Transaction Records:

- Users have the option to export their transaction records as PDF files for various purposes such as record-keeping or sharing with others.

Upon request, the server retrieves the user's transaction data and dynamically generates a PDF document using FPDFlibrary, containing a formatted representation of the transaction history.

Users can download the generated PDF file directly from the website for offline access or distribution.

Responsive Design:

Input Data Validation:

- Client-side validation is implemented using JavaScript to validate user input and provide immediate feedback on input errors (e.g., invalid date format, missing required fields) without requiring server interaction.

Server-side validation is performed to further validate and sanitize incoming data, preventing injection attacks and ensuring that only valid transactions are processed and stored in the database.

Access Controls and Session Management:

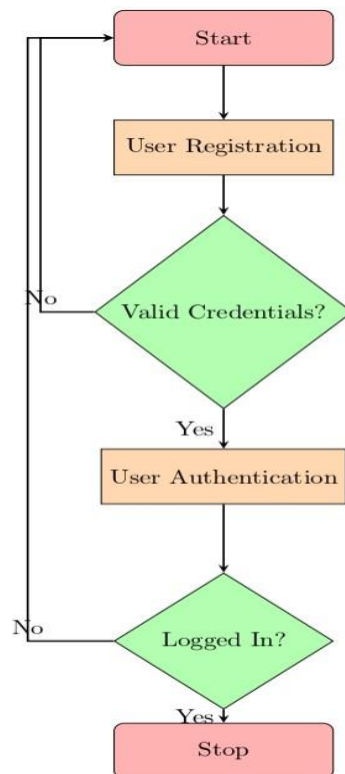
Session management mechanisms (e.g., secure cookies, server-side sessions) are implemented to maintain user sessions securely and handle user authentication throughout the browsing session.

Authentication tokens and encrypted communication (HTTPS) are used to protect user credentials and sensitive data transmitted between the client and server, ensuring data confidentiality and integrity.

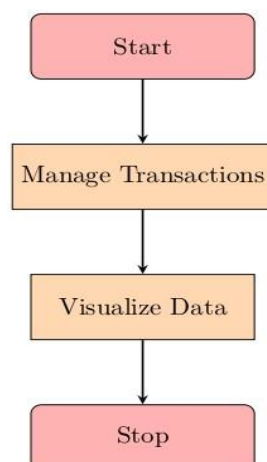
By implementing these detailed behavioral aspects, the Rupiya expense and income money manager provides users with a secure, intuitive, and feature-rich platform for effectively managing their finances and gaining valuable insights into their financial activities.

PROJECT WORKING WITH THE HELP OF DATA FLOW DIAGRAMS

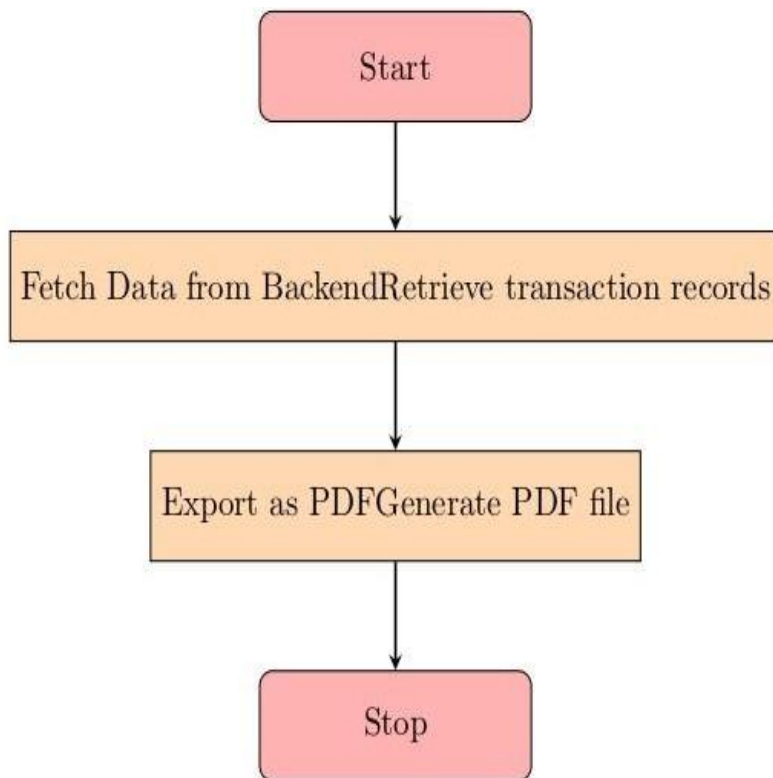
Registration of the new user:



Transaction management:



PDF export:



SOURCE CODE

LOGIN/SIGNUPPAGE:

```
<div class="header">
  <nav>
    
    <ul class="navlinks">
      <li><a href="">E-GHULAK</a></li>
      <li><a href="aboutpage.php">ABOUT US </a></li>
      <li><a href="contactus.php">CONTACT US</a> </li>
    </ul>
  </nav>
</div>
<div class="mainLoginSection">
  <div class="signIn" id="signIn">
    <form id="login" class="input-group" action="loginPage.php" method="POST">
      <h1>Login</h1>
      <div class="inputBox">
        <i class="bx bxl-gmail"></i>
        <input type="email" placeholder="email" name="email" value="{<?php echo $i ?>}">
      </div>
      <div class="inputBox">
        <i class="bx bxs-lock-alt"></i>
        <input type="password" placeholder="password" id="password" required name="password" class="form-control" value="{<?php echo $pass ?>}">
        
      </div>
      <div class="rememberForgot">
        <label><input type="checkbox" name="remember">Remember me</label><br>
      </div>
      <div class="loginbutton">
        <button type="submit" value="login" name="login" class="button">Login</button>
      </div>
    </form>
  </div>
  <div class="signUp" id="signUp">
    <form id="signup" class="input-group" action="signup.php" method="POST">
      <h1>
        Sign up
      </h1>
      <div class="inputBox">
        <i class="bx bxs-user"></i>
        <input type="text" placeholder="username" required name="username">
      </div>
      <div class="inputBox">
        <i class="bx bxs-phone"></i>
        <input type="tel" placeholder="phone number " required name="phone_number">
      </div>
      <div class="inputBox">
        <i class="bx bxl-gmail"></i>
        <input type="email" placeholder="email" required name="email">
      </div>
      <div class="inputBox">
        <i class="bx bxs-lock-alt"></i>
        <input type="password" placeholder="password" required name="password">
      </div>
      <div class="loginbutton">
        <button type="submit" class="button" value="submit" name="submit"> sign up </button>
      </div>

      <div class="create">
        <p> Already have an account ? <a href="loginPage.php"> log in </a> </p>
      </div>
    </form>
  </div>
  <div class="toggler" id="toggler">
    <h1>Hello, Friend!</h1>
    <p class="regi">Register with your personal details to use all of site features</p>
    <div class="button" id="togglerButton" onClick="moveLeftRight()">
      SIGN IN
    </div>
  </div>
</div>
</div>
```


HOME PAGE:

```


<div class="msg-container">
  <div id="slider">
    <div class="msg-col">
      <h1>
        Welcome to <br />
        E-Gulak
      </h1>
      <p>Your Money Manager!</p>
    </div>

    <div class="msg-col">
      <h1>Save Responsibly with E-Gulak</h1>
      <p>Track your income and expenses effortlessly.</p>
    </div>

    <div class="msg-col">
      <h1>Avoid Overspending</h1>
      <p>Stay on top of your finances and make informed decisions.</p>
    </div>

    <div class="msg-col">
      <h1>Interactive Chart</h1>
      <p>
        Visualize your financial data for better insights and planning.
      </p>
    </div>
  </div>
</div>
<div class="controller">
  <div id="line1"></div>
  <div id="line2"></div>
  <div id="line3"></div>
  <div id="line4"></div>
  <div id="active"></div>
</div>
<div class="features">
  <h1>Features our users love</h1>
  <div>
    <div>
      
      <h3>Save data securely</h3>
      <p>Safely save your data records on just one click </p>
    </div>
    <div>
      
      <h3>Check records </h3>
      <p>View and check your data records to maintain account </p>
    </div>
    <div>
      
      <h3>Synchronize data</h3>
      <p>Safely synchronize across devices with e-gulak.</p>
    </div>
    <div>
      
      <h3>Depth view </h3>
      <p>e-gulak provides the depth view with analytic charts</p>
    </div>
    <div>
      
      <h3>Saving plan</h3>
      <p>Keep track on savings process to meet your financial goals.</p>
    </div>
    <div>
      
      <h3>Multiple devices</h3>
      <p>Safely save your data records on just one click on multiple devices
    </div>
  </div>
</div>
```

RECORD TO PDF:

```
<?php
require_once 'fpdf/fpdf.php';
require_once 'expensesdb.php';

$sql = "SELECT * FROM expenses";
$result = mysqli_query($con, $sql);

if(isset($_POST['btn_pdf'])) {
    $pdf = new FPDF( 'P', 'mm', 'A4' );
    $pdf->AddPage();
    $pdf->SetFont('Arial', 'B', 16);
    $pdf->Cell(40, 10, 'Expense Report', 0, 1, 'C', 0);
    $pdf->Ln(10);
    $pdf->Cell(25, 10, 'Amount', 1, 0, 'C', 0);
    $pdf->Cell(40, 10, 'Date', 1, 0, 'C', 0);
    $pdf->Cell(35, 10, 'Category', 1, 1, 'C', 0);

    while ($row = mysqli_fetch_assoc($result)) {
        $pdf->Cell(25, 10, $row['amount'], 1, 0, 'C', 0);
        $pdf->Cell(40, 10, $row['date'], 1, 0, 'C', 0);
        $pdf->Cell(35, 10, $row['category'], 1, 1, 'C', 0);
    }

    // Output the PDF
    $pdf->Output();
}
?>
```

ANALYSIS OF ACCOUNT:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "income";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT date , amount FROM income ORDER BY date";
$result = $conn->query($sql);

$dataPoints = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $date = strtotime($row["date"]) * 1000;
        $dataPoints[] = array("x" => $date, "y" => $row["amount"]);
    }
} else {
    echo "0 results";
}

// Close connection
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Analysis Page</title>
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <!-- <link rel="stylesheet" href="styles.css" -->
    <link rel="stylesheet" href="AnalysisPage.css">
    <link href="https://fonts.googleapis.com/css2?family=Rubik:wght@400;700&display=swap" rel="stylesheet" />
    <link href="https://unpkg.com/bonicons@2.1.4/css/bonicons.min.css" rel="stylesheet">
    <script>
        window.onload = function() {
            var chart = new CanvasJS.Chart("chartContainer", {
                animationEnabled: true,
                title: {
                    text: "Income"
                },
                axisX: {
                    title: "Date",
                    valueFormatString: " DD MM YY",
                },
                axisY: {
                    title: "Income (in rupees)",
                    valueFormatString: "$,###",
                    prefix: "₹"
                },
                data: [
                    {
                        type: "spline",
                        markersize: 5,
                        xValueType: "datetime",
                        dataPoints: <?php echo json_encode($dataPoints, JSON_NUMERIC_CHECK); ?>
                    }
                ]
            });
            chart.render();
        }
    </script>
```

ACCOUNTS TRANSACTION RECORDS:

```
<form id="pdf-form" action="pdf_gen.php" method="POST" name="btn_pdf" >
<button type="submit" name="btn_pdf">Get PDF</button>
</form>

<h1>Expense and Income record </h1>

<table id="expense-table">

  <tr>

    <th>amount</th>
    <th>date</th>
    <th>category</th>
    <th>operation</th>

  </tr>

  <?php
$sql = "SELECT * FROM expenses LIMIT 5"; // Display only the first 10 records
$result = mysqli_query($con, $sql);
if ($result) {
  while ($row = mysqli_fetch_assoc($result)) {
    $amount = $row['amount'];
    $date = $row['date'];
    $category = $row['category'];
    echo <tr>

      <td> - ₹ ' . $amount . '</td>
      <td>' . $date . '</td>
      <td>' . $category . '</td>
      <td>
        <div class="button-container">
          <button><a href="update.php?updateid=' . $row['id'] . '">update</a></button>
          <button><a href="delete.php?deleteid=' . $row['id'] . '">delete</a></button>
        </div>
      </td>

    </tr>;
  }
}
?>

<?php
include 'incomedb.php';
if(isset($_POST['submit'])) {
  $amount = $_POST['amount'];
  $date = $_POST['date'];
  $category = $_POST['category'];
  $sql = "INSERT INTO income (amount, date, category) VALUES ('$amount', '$date', '$category')";
  $result = mysqli_query($con, $sql);
  if($result) {
    header('location:record.php'); }
  else {
    die(mysqli_error($con));
  }
}
?>

<?php
$sql = "SELECT * FROM income LIMIT 5"; // Display only the first 10 records

$result = mysqli_query($con, $sql);
if ($result) {
  while ($row = mysqli_fetch_assoc($result)) {

    $amount = $row['amount'];
    $date = $row['date'];
    $category = $row['category'];
    echo <tr>
```

ADDING INCOME AND EXPENSES INTO ACCOUNT:

```
<?php

$host = 'localhost';
$db   = 'expenses';
$user = 'root';
$pass = '';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES  => false,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_expense'])) {
    $amount = $_POST['amount'];
    $date = $_POST['date'];
    $category = $_POST['category'];

    $sql = "INSERT INTO expenses (amount, date, category) VALUES (?, ?, ?)";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$amount, $date, $category]);
    header("Location: {$_SERVER['PHP_SELF']}");
    exit;
}

$sql = "SELECT * FROM expenses";
$stmt = $pdo->query($sql);

?>
```

TESTINGANDIMPLEMENTATION

Testing:

Element Testing: Individual components of the system, such as functions, modules, and classes, are tested in isolation to ensure they function correctly.

Integration Testing: Different modules and components are integrated and tested together to verify their interactions and compatibility.

System Testing: The entire system is tested as a whole to validate its behaviour and functionality according to the requirements

User Acceptance Testing (UAT): End-users participate in testing the system to ensure it meets their needs and expectations. Feedback from UAT helps identify any usability issues or bugs that need to be addressed.

Performance Testing: The system's performance is evaluated under various conditions, such as different user loads and network conditions, to ensure It meets performance requirements and can handle expected traffic levels.

Security Testing: Security measures, such as authentication, authorization, data encryption, and secure communication protocols, are tested to identify and mitigate potential vulnerabilities and threats.

Compatibility Testing: The system is tested across different devices, browsers, and operating systems to ensure it works correctly and displays properly on all platforms.

Implementation:

Data Migration: If applicable, existing data may be migrated to the new system, ensuring continuity and preserving user data.

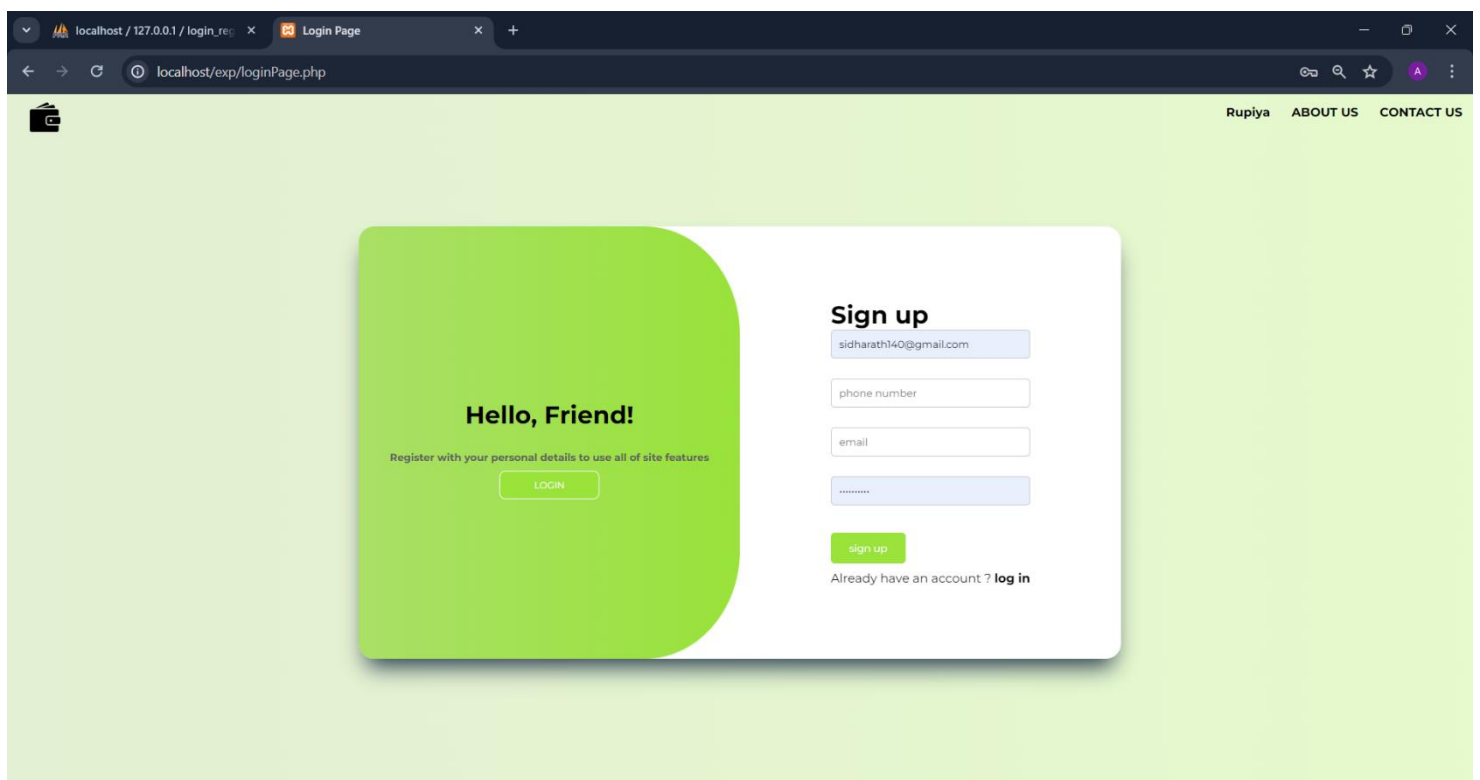
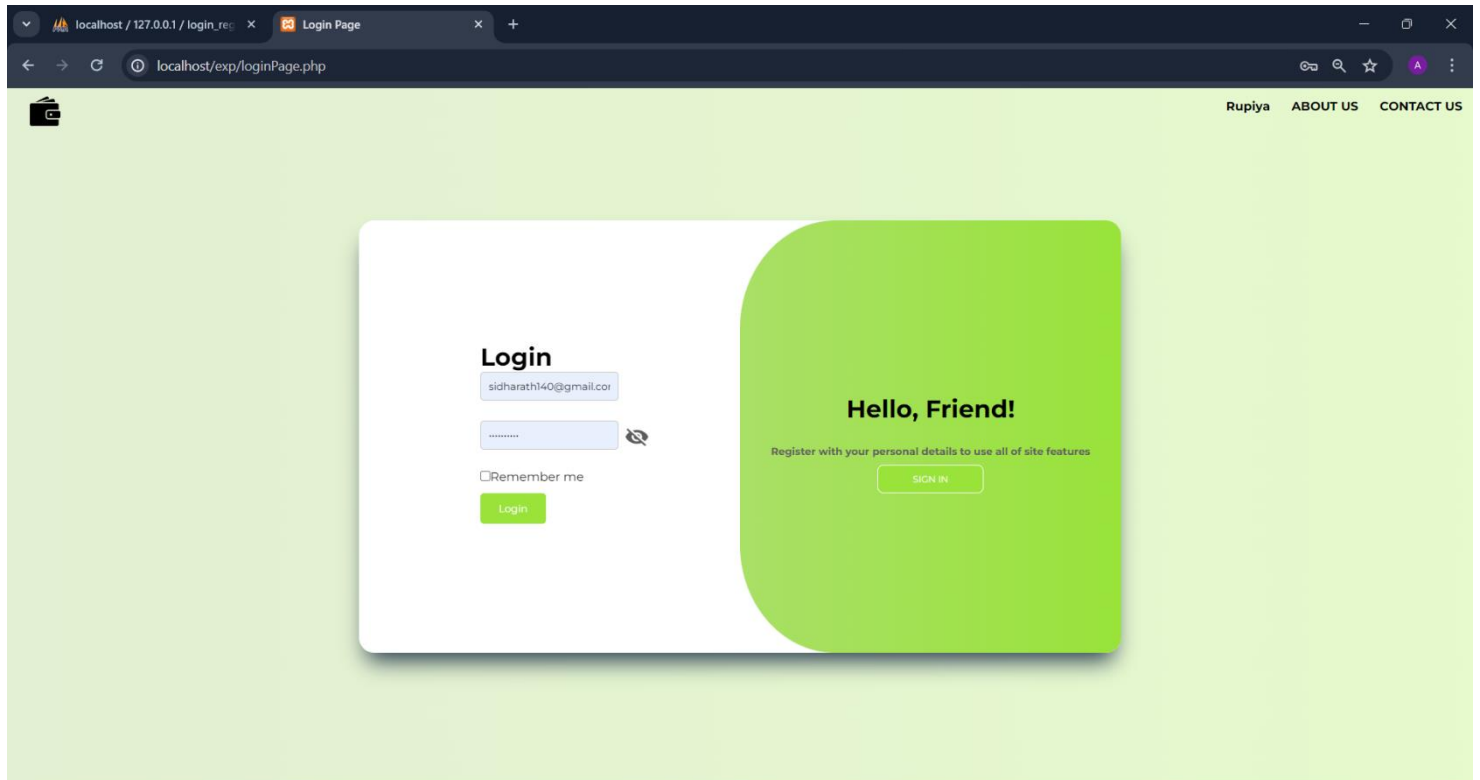
Monitoring and Maintenance: Continuous monitoring of the system's performance, security, and uptime is essential to ensure smooth operation. Regular maintenance and updates are performed to address bugs, security vulnerabilities, and feature enhancements.

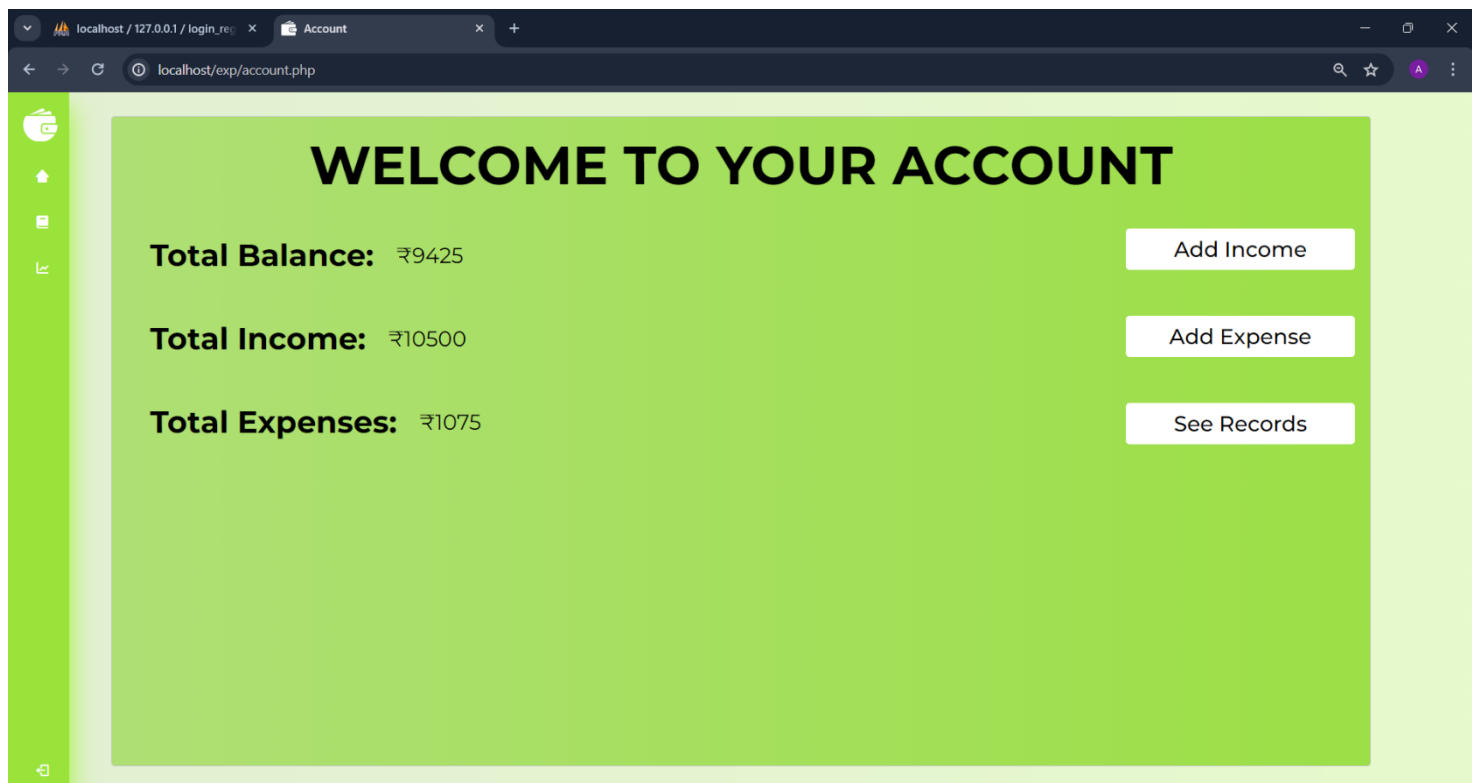
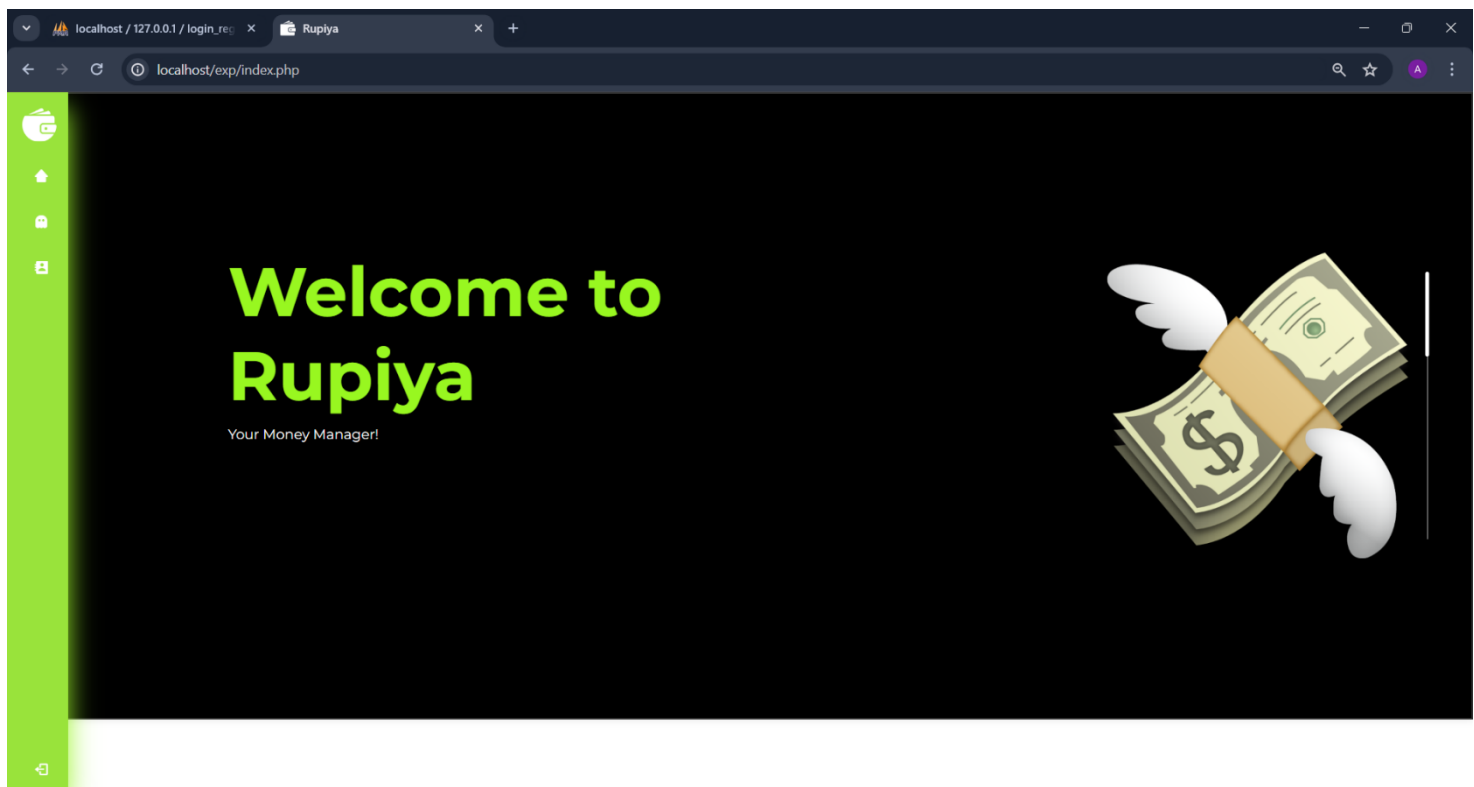
User Support: A system for providing user support, such as a help desk or knowledgebase is established to assist users with any issues or questions they may have.

Feedback Mechanism: A feedback mechanism is implemented to collect user feedback and suggestions for improving the system. This feedback is valuable for making iterative improvements and enhancements to the system over time.

By conducting thorough testing and following a systematic implementation process, the project ensures that the system is reliable, functional, and user-friendly, meeting the needs and expectations of its users. Ongoing monitoring, maintenance, and support further contribute to the successful operation of the system post-implementation.

SNAPSHOT OF PROJECT(WORKING)





Add Income

Amount:

Date:

Category:

Add Income

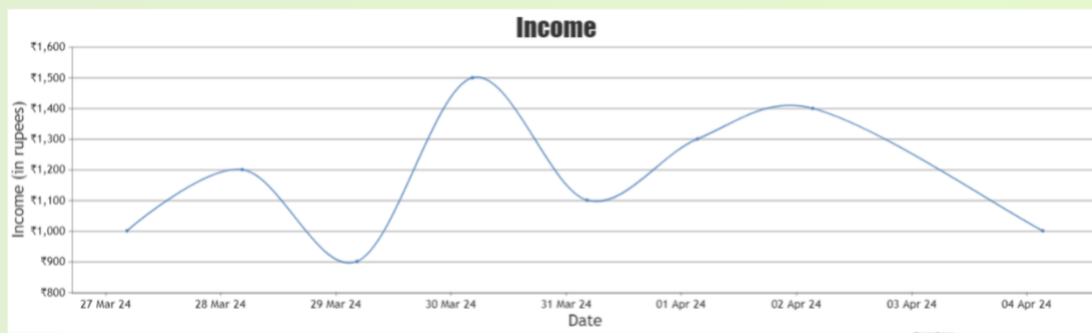
Expense and Income record

Get PDF

amount	date	category	operation
- ₹ 20	2024-04-03	salary	<div><div>update</div><div>delete</div></div>
- ₹ 300	2024-04-03	food	<div><div>update</div><div>delete</div></div>
- ₹ 100	2024-04-02	food	<div><div>update</div><div>delete</div></div>
- ₹ 100	2024-03-27	Groceries	<div><div>update</div><div>delete</div></div>
- ₹ 50	2024-03-28	Utilities	<div><div>update</div><div>delete</div></div>
+ ₹ 1000	2024-04-04	salary	<div><div>update</div><div>delete</div></div>
+ ₹ 1000	2024-03-27	Income	<div><div>update</div><div>delete</div></div>
+ ₹ 1200	2024-03-28	Income	<div><div>update</div><div>delete</div></div>
+ ₹ 900	2024-03-29	Income	<div><div>update</div><div>delete</div></div>
+ ₹ 1500	2024-03-30	Income	<div><div>update</div><div>delete</div></div>

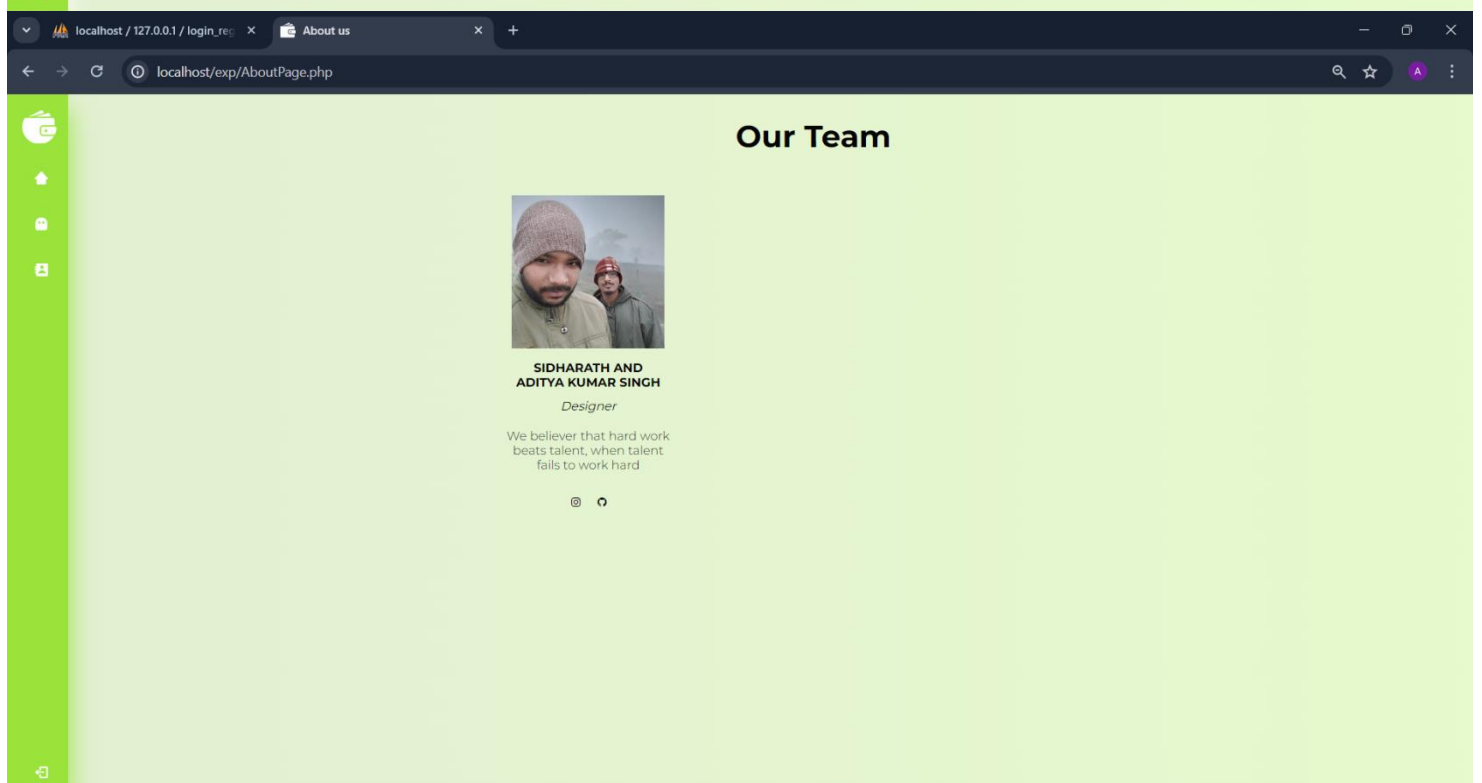
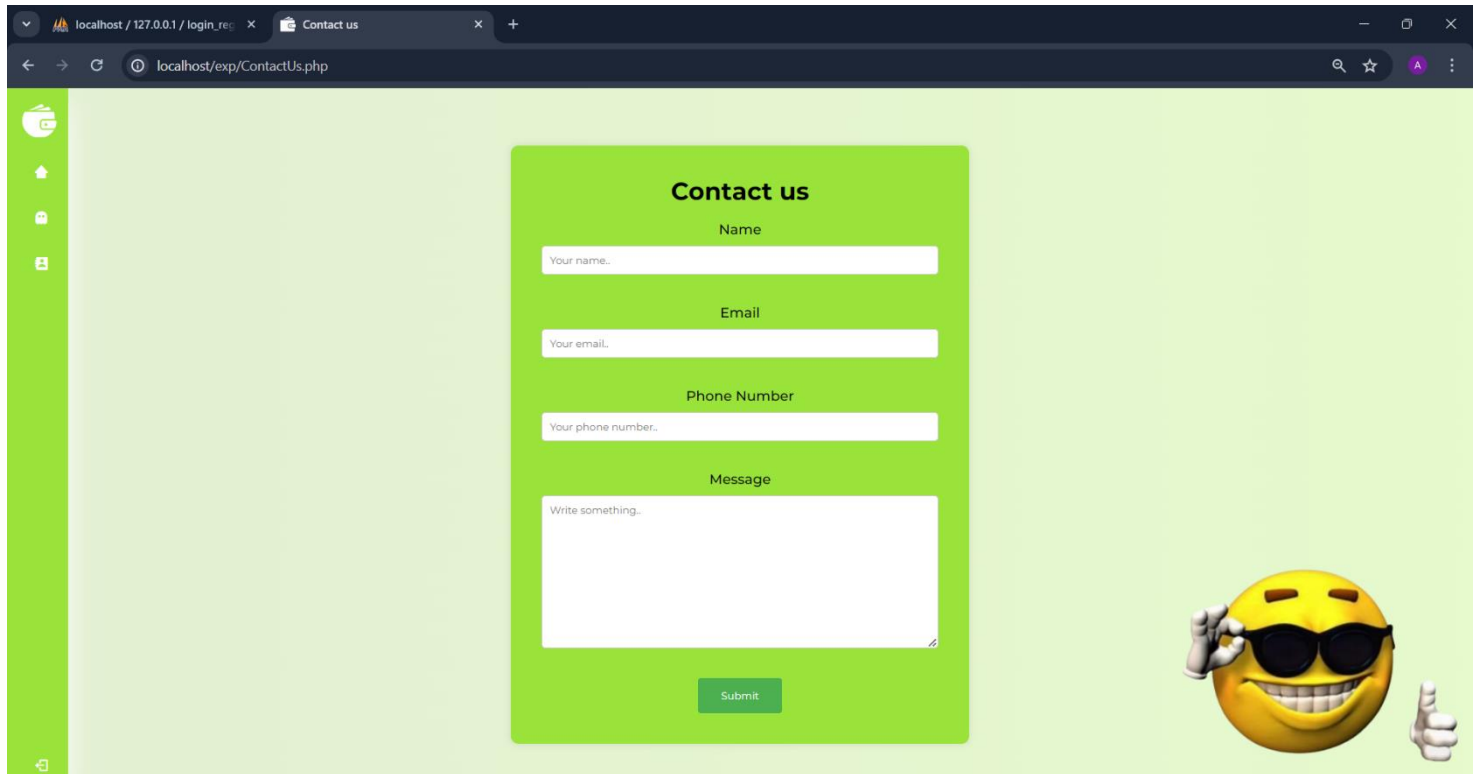


Expense



Income





CONCLUSION

RUPIYA offers a comprehensive suite of financial management tools with features including budgeting, expense tracking, and reporting. Its user-friendly interface caters to both novice and experienced users, making it accessible to a wide audience. Security measures such as encryption protocols ensure the protection of users' financial data. The active developer community provides ongoing support and frequent updates, enhancing the software's reliability and functionality. With its cross-platform compatibility and positive user feedback, RUPIYA emerges as a reliable choice for personal finance management.

REFERENCES

1. W3Schools
2. GeeksforGeek
3. YouTube

Links:

1. <https://www.w3schools.com>
2. <https://www.geeksforgeeks.org>
3. <https://www.youtube.com>